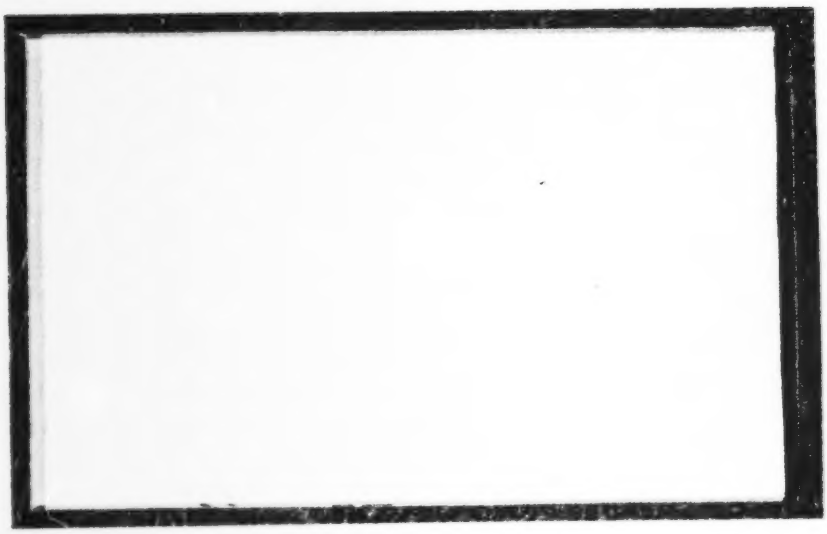
UNIVERSITÉ DE MONTRÉAL



CENTRE DE RECHERCHE SUR LES TRANSPORTS

CASE POSTALE 6128, SUCCURSALE A, MONTRÉAL, QUÉBEC, H3C 3J7

OPTIMAL TOUR PLANNING WITH SPECIFIED NODES

by

Gilbert Laporte*, Hélène Mercure**
and Yves Nobert**

The authors are grateful to the Canadian Natural Sciences and Engineering Research Council (grant A4747) and to the Québec Government (F.C.A.C. grant 83EQ0428) for their financial support.

* Ecole des Hautes Etudes Commerciales de Montréal, 5255, av. Decelles, Montréal H3T 1V6

** Département des Sciences administratives, Université du Québec à Montréal, 1495, rue Saint-Denis, Montréal H3C 3P8

Université de Montréal

Centre de recherche sur les transports - Publication #330
Ecole des Hautes Etudes Commerciales
Les Cahiers du Gérad - G-83-14

Juin 1983

Dépôt légal - 1er trimestre 1984

ABSTRACT

This paper consider the problem of determining the shortest circuit or cycle in a graph containing n nodes and such that (i) each of k nodes ($k \leq n$) is visited exactly once; (ii) each of the remaining n-k nodes is visited at most once. A branch and bound algorithm for this problem is described. Results are presented for problems involving up to 200 nodes in the asymmetrical case and up to 80 nodes in the symmetrical case.

Key words: travelling salesman problem, specified nodes, branch and bound.

RESUME

On considère le problème consistant à déterminer le circuit ou cycle le plus court dans un graphe contenant n noeuds et de telle sorte que (i) k noeuds ($k \leq n$) soient visités chacun une et une seule fois; (ii) chacun des n-k noeuds restants soit visité au plus une fois. On décrit un algorithme de "branch and bound" pour ce problème et on présente des résultats numériques pour des problèmes asymétriques contenant jusqu'à 200 noeuds et des problèmes symétriques contenant jusqu'à 80 noeuds.

Mots clés: problème du voyageur de commerce, noeuds spécifiés, "branch and bound".

1. INTRODUCTION

In its most common interpretation, the travelling salesman problem (TSP) consists of determining the shortest route for a salesman wishing to visit each of n cities once and only once [3]. Over the last three decades, this problem has attracted the attention of several operational researchers and has led to some significant developments in the O.R. field [6]. It is well known however that in its pure form, the TSP seldom fits the routing problems really encountered by salesmen since in practice, some of the restrictions of the TSP may be lifted (see for example [11]) while extra constraints may have to be incorporated (see for example [8,10]).

The following problem studied by Saksena and Kumar [13], Dreyfus [4] and Ibaraki [7] applies to situations where only a subset of the n cities have to be visited by the salesman. More specifically, let $N = \{1,\ldots,n\}$ be a set of nodes (cities), $K = \{1,\ldots,k\}$ $(k \leq n)$ a set of "specified" nodes (i.e. those requiring a visit) and $C = (c_{ij})$ a non-negative distance matrix defined on N^2. Each pair (i,j) defines an arc from i to j.

We wish to determine the shortest circuit passing through each node of K exactly once; we shall refer to this problem as the STSP (TSP with specified nodes). As was shown in [9], the difficulty of the problem depends largely on the nature of C and on the degree imposed on nodes in N-K:

(i) if C satisfies the triangle inequality (i.e. if $c_{ij} \leq c_{ik} + c_{kj}$ $(i,j,k \in N)$), the STSP always reduces to
- a TSP on N if the degree of nodes in N-K equals 2 or if it is greater than or equal to 2;
- a TSP on K if the degree of nodes in N-K is less than or equal to 2 or unspecified.

Therefore the problem presents little interest in this case.

(ii) if C does not satisfy the triangle inequality, the only interesting case is that where the degree of nodes in N-K must be at most 2 (other cases also reduce to the TSP or to an easier problem). This short note proposes an efficient algorithm for the solution of this problem.

2. IBARAKI'S ALGORITHM

The problem considered by Saksena and Kumar [13], Dreyfus [4] and Ibaraki [7] is that of determining the shortest path between a source and a sink, passing through a set of specified nodes exactly once and through the other nodes at most once. It can be shown that this problem is very similar to the STSP treated in this paper.

The following formulation for the STSP can be derived from Ibaraki's paper: let us define a binary variable x_{ij} as follows:

(i) if $i \neq j$, x_{ij} indicates the presence ($x_{ij} = 1$) or the absence ($x_{ij} = 0$) of an arc from node i to node j in the optimal solution;

(ii) if $i = j$, x_{ij} indicates whether node i is used ($x_{ij} = 0$) or not ($x_{ij} = 1$) in the optimal solution.

The problem is then to

(P) minimize $\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$

subject to

(1) $\sum_{i=1}^{n} x_{ij} = 1 \qquad (j \in N)$

$$(2) \qquad \sum_{j=1}^{n} x_{ij} = 1 \qquad (i \in N)$$

$$(3) \qquad \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} \leq |S| - 1 \qquad (S \subset N,\ S \cap K \neq K,\ S \cap K \neq \phi)$$

$$(4) \qquad x_{ij} = 0 \text{ or } 1 \qquad (i,j \in N)$$

In this formulation, c_{ij} is set equal to

(i) 0 if $i \in N-K$;

(ii) an arbitrary large number M if $i \in K$.

This ensures that x_{ij} will be equal to zero if $i \in K$; therefore all nodes belonging to K will be used in the optimal solution.

Constraints (1), (2) and (4) require no explanations. Constraints (3) are imposed in order to eliminate subtours containing some but not all nodes of K: these are clearly illegal since the optimal tour must contain all nodes of K. On the other hand, it is not necessary to consider cases where $S \cap K = \phi$ since it is never advantageous (in terms of the objective function) to produce subtours disconnected from K.

It can be seen that

(i) if $K = \phi$, (i.e. constraints (3) are removed), (P) reduces to an assignment problem for which there exist efficient algorithms [1];

(ii) if K = N, this formulation is identical to that of the asymmetrical TSP (see [3] for example);

(iii) otherwise, the difficulty of the STSP should lie somewhere between that of the assignment problem and that of the TSP.

The algorithm proposed by Ibaraki for the STSP is similar to the branch and bound method developed by Shapiro [14] for the TSP. It can be summarized as follows.

(i) At each node h of the search tree, we define E_h, the set of arcs excluded from the solution (at the first node of the tree, $E_h = \phi$). An assignment problem constrained by E_h is solved.

(ii) Consider the subtours contained in the solution at node h. If there is only one subtour, it constitutes a feasible solution and a backtracking procedure is applied. Otherwise, consider the subtour with the minimum number of arcs. This subtour is characterized by a set of nodes $(r_1,\ldots,r_m)$ and a set of arcs $((r_1,r_2),\ldots,(r_m,r_1))$. To the descendant nodes j from node h are associated the following sets of excluded arcs:

$$E_j = E_h \cup ((r_j, r_{j+1}))$$

where $r_{m+1} = r_1$.

(iii) Branching is always made on the pending node having the least lower bound for the STSP.

(iv) The procedure ends when all branches of the tree have been explored, according to the usual branch and bound rules.

Ibaraki reports computational results for a limited number of randomly generated problems involving 21 and 31 nodes, on the Kyoto University FACOM 230-60 computer. The assignment problems were solved by means of Munkres' algorithm [12].

6. [illegible]

The main attraction of Gerstel's algorithm (as in [illegible]) fact that at each node of the search tree, no probe since it an assignment problem, and therefore, no shortest [illegible] during the whole process. Furthermore, the algorithm only [illegible] the fact that the SSP is a relaxation of the TP, in that [illegible] hours than in the TSP need to be matched pairs.

However, Gerstel's results can be improved by using [illegible] tage of recent developments in the construction of algorithms for the assignment problem [4] and for the TSP (see for example [13]). Further, we feel that additional computational tests are required to validate the suggested approach.

In their paper on the TSP, Grossetti and Toth demonstrated that Gerstel's algorithm can be easily improved by using a more efficient algorithm for the assignment problem [4] and by defining the rule for generating subproblems. They use the partitioning scheme proposed by Garfinkel [3] to which they add a refinement. At each node k of the tree, S_k is defined as above and I_k is the set of arcs included in the solution (at the first node of the tree, $I_k = \emptyset$). Branching is made from the subset with the number of arcs included in I_k (as opposed to the subset with the minimum number of arcs as in [14] and in [13]). S_k is defined as in [14] and I_k as in [13]:

$$I_k = \begin{cases} I_p & \text{if } k = 1 \\ I_p \cup \{(i_p, i_{p+1})\} : \ q = \max\{j : (i_j, i_{j+1}) \in I_p\} & \text{if } k > 1 \end{cases}$$

The overall effect of this strategy is to drastically reduce the number of nodes in the search tree. The authors report results for TSP's ranging from 60 to 200 cities.

We therefore suggest applying a similar approach to the STSP: i.e. we use the same relaxation as Ibaraki but the approach developed by Carpaneto and Toth for the solution of the assignment problems and for generation of the subproblems.

4. COMPUTATIONAL RESULTS

It is now well known that the computational performance of subtour elimination algorithms such as those described by Shapiro [10] and by Carpaneto and Toth [2] varies greatly according to whether C is symmetrical (i.e. $c_{ij} = c_{ji}$ for all $i,j \in N$) or not. In the fist case, this type of approach is less efficient since a large number of subtours involving only two nodes are generated and have to be eliminated; this phenomenon is far less frequent in asymmetrical problems. We have tested the algorithm on both types of problems.

The algorithm was first tested on a series of asymmetrical problems ranging from 80 to 200 nodes. The c_{ij}'s for these problems were randomly generated from a uniform distribution on [0,100]. For each value of n, k was successively set at n/4, n/2, 3n/4 and n. Five problems of each type were solved; table 1 reports average values. All problems were solved on the University of Montreal Cyber 173 computer, using an FTN4 compiler. Memory appears to be the main factor limiting the size of the largest problems which could be solved by the algorithm: all problems involving no more than 200 nodes could be solved within 20 seconds; however, problems containing more than 200 nodes required more than the maximum memory allowed ($200\ 000_8$ words).

For a given value n, we observe that the time required to solve the problem is not monotonic with respect to k: there appears to be a peak at about n/2 and a trough at 3n/4. (The peak

was noted by Ibaraki [7], but not the trough.) These results were confirmed by solving small problems (n = 20 and 30) with all values of k from 1 to n.

TABLE 1

Results for asymmetrical problems
(average values on 5 random problems)

n		k = n/4	k = n/2	k = 3n/4	k = n
80	Time	1.65	3.52	3.34	3.35
	AP (1)	9.0	28.0	33.8	41.0
	Q (2)	5.6	10.8	15.2	15.6
	Nodes (3)	3.0	4.4	5.8	6.6
120	Time	5.96	6.69	3.71	5.79
	AP	25.0	30.6	20.8	36.0
	Q	11.4	15.4	11.0	27.6
	Nodes	5.2	5.6	5.2	7.4
160	Time	8.52	10.9	6.66	6.59
	AP	24.8	39.8	56.4	57.2
	Q	10.0	22.4	11.4	16.2
	Nodes	6.2	8.6	6.2	5.6
200	Time	8.70	18.19	12.16	15.31
	AP	15.4	53.0	60.0	128.0
	Q	8.6	26.8	23.8	35.8
	Nodes	2.6	7.4	7.8	13.4

(1) AP: number of assignment problems solved.
(2) Q: number of subproblems inserted in the queue.
(3) Nodes: number of nodes explored in the branch and bound tree.

The algorithm was also tested on symmetrical problems not satisfying the triangle inequality. These were generated as in the asymmetrical case with the additional requirement that $c_{ij} = c_{ji}$. As expected, a very large number of subtours involving only two nodes were observed, leading to excessive memory requirements; computa-

tion times were also larger than in the asymmetrical cases but in general better than those reported in an earlier paper on the symmetrical STSP [9]. Results for problems ranging from 20 to 80 cities are reported in table 2.

TABLE 2

Results for symmetrical problems
(average values on the number of successful problems out of 5(1))

n		k = n/5	k = 2n/5	k = 3n/5	k = 4n/5	k = n
20	Time	0.09	0.19	0.54	1.81	4.56
	AP (2)	6.8	10.2	25.2	77.2	179.6
	Q (3)	4.0	6.4	19.0	45.0	106.4
	Nodes (4)	3.8	5.8	13.4	38.6	88.6
40	Time	0.50	4.07	23.72	21.65(4)	(0)
	AP	18.2	79.2	390.6	315.2	
	Q	10.6	53.0	226.8	256.5	
	Nodes	10.4	39.2	197.8	162.0	
60	Time	2.91	23.62	(0)	(0)	(0)
	AP	49.6	203.0			
	Q	27.2	140.0			
	Nodes	25.4	103.2			
80	Time	14.8	(0)	(0)	(0)	(0)
	AP	145.2				
	Q	85.6				
	Nodes	77.0				

(1) Whenever the number of successful problems is less than 5, this is indicated in brackets.
(2) AP: number of assignment problems solved.
(3) Q: number of subproblems inserted in the queue.
(4) Nodes: number of nodes explored in the branch and bound tree.

BIBLIOGRAPHY

[1] CARPANETO, G., TOTH, P.: "Solution of the Assignment Problem", ACM Transactions on Mathematical Software, vol.6, no 1, pp.104-111, 1980.

[2] CARPANETO, G., TOTH, P.: "Some New Branching and Bounding Criteria for the Asymmetric Travelling Salesman Problem", Management Science, vol.26, no 7, pp.736-743, 1980.

[3] DANTZIG, G.B., FULKERSON, D.R., JOHNSON, S.M.: "Solution of a Large Scale Travelling Salesman Problem", Operations Research, vol.2, pp.393-410, 1954.

[4] DREYFUS, S.E.: "An Appraisal of Some Shortest-Path Algorithms", Operations Research, vol.17, pp.395-412, 1969.

[5] GARFINKEL, R.S.: "On Partitioning the Feasible Set in a Branch-and-Bound Algorithm for the Asymmetric Travelling Salesman Problem", Operations Research, vol.21, pp.340-343, 1973.

[6] HELD, M.: "Influences of the Travelling Salesman Problem", Paper presented at the XI. International Symposium on Mathematical Programming, Bonn, 1982.

[7] IBARAKI, T.: "Shortest Path Problems Visiting Specified Nodes", Electronics and Communications in Japan, vol.53-A, pp.10-18, 1970.

[8] LAPORTE, G., NOBERT, Y.: "A Branch and Bound Algorithm for the Capacitated Vehicle Routing Problem", to appear in Operations Research Spektrum, 1983.

[9] LAPORTE, G., NOBERT, Y.: "Finding the Shortest Cycle through k Specified Nodes", Cahiers du GERAD G-83-07, Ecole des Hautes Etudes Commerciales de Montréal, 11 pages, 1983.

[10] LAPORTE, G., NOBERT, Y., DESROCHERS, M.: "Two Exact Algorithms for the Distance Constrained Vehicle Routing Problem", to appear in Networks, 1984.

[11] MILIOTIS, P., LAPORTE, G., NOBERT, Y.: "Computational Comparison of Two Methods for Finding the Shortest Complete Cycle or Circuit in a Graph", RAIRO (recherche opérationnelle), vol.15, pp.233-239, 1981.

[12] MUNKRES, J.: "Algorithms for the Assignment and Transportation Problems", Journal of SIAM, vol.5, no 1, p.32, 1957.

[13] SAKSENA, J.P., KUMAR, S.: "The Routing Problem with "K" Specified Nodes", Operations Research, vol.14, pp.909-913, 1966.

[14] SHAPIRO, D.M.: "Algorithms for the Solution of the Optimal Cost and Bottleneck Travelling Salesman Problems". Ph.D. Thesis, Washington University, 1966.

POOR

QUALITY

ORIGINAL